

  برنامه سازی پیشرفته

پاییز 1392

دانشگاه زنجان

بازی نبرد هوایی

مقدمه:

در این پروژه شما باید ، با پیاده سازی کلاس های مختلف ، یک جنگ هوایی را پیاده سازی کنید.کلاس های این پروژه بر اساس حداقل امکانات جنگ جهانی دوم در نبرد های هوایی است.در این راهنما ، تعدادی کلاس (که پایه بازی است) ، با شرح کامل ذکر شده است.طبیعی است که این کلاس های توانایی نمایش یک جنگ تمام عیار هوایی را ندارد.پس انتظار می رود که شما هم ، با خلاقیت ، این بازی را جذاب تر کنید.[[1]](#footnote-1)

داستان و امکانات بازی:

ابتدا باید بدانیم روند بازی به چه شکلی است.در یک نبرد هوایی تعدادی هواپیما با یکدیگر مبارزه می کنند. هر هواپیما به یک ملیت وابسته است.[[2]](#footnote-2)هدف هر ملیت برای جنگ ، از بین بردن هواپیمای دیگر ملیت هاست.پس بازی تا زمانی ادامه دارد که فقط یک هواپیما در بازی بماند.

در آسمان وضعیت هوایی ممکن است ثابت نباشد. آسمان می تواند ، صاف و یا مه آلود باشد.در هوای مه آلود خلبان ترجیح می دهد با توجه به قطب نمای موجود در هواپیما اول از مه خارج شود.پس ، بعد از ورود به مه باید به طور مستقیم ، سریع و در جهت وارد شدن از آن خارج شود.

علاوه بر آسمان ، زمین هم امن نیست ! ضد هوایی ها ، می توانند تمام هواپیما هایی که در محدوده بردشان است را از بین ببرند. مگر اینکه به وسیله بمب از بین بروند. این ضد هوایی ها ، به هیچ ملیتی وابسته نیستند.پس هیچ هواپیمایی از دستشان در امان نیست.

یکی از موانع موجود در آسمان شهر ها در زمان جنگ ، کشتی هوایی است. این موانع به هیچ کسی آسیب نمی رسانند ولی فقط به جهت سد کردن راه هواپیما ها ، محکوم به از بین رفتن اند.هواپیما ها با شلیک موشک می توانند این موانع را از بین ببرند.



همیشه در نبرد های هوایی ، موانع طبیعی وجود دارند.یکی از موانع طبیعی ، کوهستان مرتفع است.این کوه ها از بین نمی روند و برخورد با آنها موجب از بین رفتن هواپیما میشود.

هر هواپیما دو نوع سلاح مختلف دارد.اولی برای از بین بردن هواپیما های دشمن و کشتی های هوایی است(موشک) و دومی برای از بین بردن ضد هوایی ها است(بمب های سقوط آزاد).

موشک ها بعد از شلیک ، به طور مستقیم و در جهت شلیک حرکت می کنند و با برخورد به موانع ، آنها را ازبین می برند.بمب های سقوط آزاد ، بعد از رها شدن ، با توجه جهت هواپیما ، در خانه بعدی می افتند و منفجر می شوند.[[3]](#footnote-3)

کلاس دیاگرام ها و توضیحات هر کلاس:

Aircraft

Name:String -

X:int -

Y:int -

isAlive:bool -

dir:char -

Aircraft(int x,int y,String name) +

goUP():bool +

goDOWN():bool +

goRIGHT():bool +

goLEFT():bool +

fire(Weapon Weapon\_type): void +

getDir():char +

kill():void +

toString():String +

1. کلاس هواپیما:

این کلاس شامل مختصات هواپیما در نقشه،جهت گیری آن و همچنین وضعیت سلامت هواپیما است.

سازنده ی این کلاس مختصات اولیه هواپیما به همراه نام آن(که می تواند نام بازیکن هواپیما باشد) را دریافت می کند.[[4]](#footnote-4)توابع حرکت هواپیما،به صورت تک حرکتی اند.یعنی فقط حرکت در چهار جهت اصلی به میزان یک خانه مجاز است.این توابع نتیجه حرکت کردن را بر می گردانند و در زمانی که حرکت خواسته شده مقدور نباشد مقدار falseرا گزارش خواهند کرد.برای شلیک از تابع fire استفاده می شود.آرگومان این تابع از نوع Weapon است که بعداً بررسی خواهیم کرد.تابع باید ، اصابت گلوله و یا بمب به هدف خواص را گزارش کند.[[5]](#footnote-5)تابع getDir جهت هواپیما را مشخص می کند.توجه کنید که جهت فعلی هواپیما به وسیله توابع حرکت تعیین می شوند.هواپیما، به وسیله ی تابع kill() (که isAlive را falseمی کند)،از بین می روند.[[6]](#footnote-6)تابع toString() مختصات فعلی هواپیما ، جهت آن و سلامت آن را در قالب String گزارش میکند.[[7]](#footnote-7)

1. کلاس خانه:

*Block*

X:int -

Y:int -

name:char -

Block() +

Block(int x,int y,char name) +

getX():int +

getY():int +

setX():int +

setY():int +

*getName():char-Abstract +*

*toString():String-Abstract +*

برای ساخت نقشه ی بازی ، جدولی به تعداد مورد نیاز ایجاد می کنیم و خانه های این جدول را با Blockپر می کنیم.کلاس Block یک کلاس Abstract است و انواع مختلف خانه ها به وسیله ی این کلاس ساخته می شوند(کلاس هوای مه آلود،ضد هوایی و...).

کلاس های ساخته شده از روی این کلاس،باید نام خاصی را داشته باشند.برای مثال کلاس ضد هوایی با کارکتر’G’ (Gun)نمایش داده میشود.تابع toString نام کلاس و مختصات آن را در قالب String برمیگرداند.دو تابع getName() و toString()،Abstract اند و در کلاس هایی که از این کلاس ارث می برند ، باید پیاده سازی شوند.[[8]](#footnote-8)

* 1. کلاس خانه ی مه آلود(Fog):هواپیما بعد از ورود به این خانه ، بلافاصله از خانه ی مقابل آن خارج می شود.[[9]](#footnote-9)
  2. کلاس هوای معمولی(Clear)
  3. کلاس ضد هوایی(AntiAirCraftGun):ورود به این خانه باعث حذف هواپیما می شود.مگر اینکه با بمب سقوط آزاد از بین برود و تبدیل به خانه ی Clear بشود.
  4. کلاس کشتی هوایی(AirShip)[[10]](#footnote-10)
  5. کلاس کوه(Mountain)
  6. کلاس خانه تصادفی(Random)[[11]](#footnote-11)

تمامی کلاس های فوق از کلاس Block ارث می برند.

کلاسRandom ، به طور اتفاقی یکی از پنج کلاس دیگر را انتخاب می کند و رفتاری شبیه با آن ، نشان می دهد.

حتماً تمام کلاس های Block را در یک Package مجزا نگه دارید.

1. کلاس سلاح:

*Weapon*

Luncher:AirCraft -

dir:char -

Weapon() +

Weapon (AirCraft Luncher) +

getDir():char +

این کلاس،شامل دو خصوصیت شلیک کننده و جهت است.جهت در این کلاس به وسیله ی جهت هواپیما در لحظه شلیک بدست می آید.

* 1. Missile
  2. bomb

دو کلاس موشک و بمب از کلاس سلاح ارث می برند.

موشک به صورت مستقیم و در هوا حرکت می کند.پس می تواند ، کشتی هوایی و هواپیماهای دشمن را نابود کند.

|  |  |  |  |
| --- | --- | --- | --- |
|  | "خانۀ بعدی" | |  |
|  | خانه فعلی هواپیما | جهت:  ↑ |  |
|  |  | |  |

بمب به صورت پرتابی است و در خانه بعدی هواپیما به زمین می خورد.پس می تواند ضد هوایی را نابود کند.(در مثال های زیر ، "خانه ی بعدی" مشخص شده است.)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
| "خانۀ بعدی" | خانه فعلی هواپیما | جهت:  → |  |
|  |  | |  |

1. کلاس نقشه:

Map

Blocks:Block[][] -

Rows:int -

Cols:int -

name:String -

Map() +

Map(int row,int col) +

Map(int row,int col,string name) +

editBlock(int row,int col,Block block):void +

getRows():int +

getCols():int +

reSetMap() +

reSetMap(int row,int col) +

toString():String +

صفحه بازی به وسیله ی این کلاس ، مشخص می شود.در این کلاس ، جدولی به اندازه دلخواه و از نوع کلاس Block ساخته می شود.خصوصیات اصلی این کلاس سطر و ستون ها ست.

سازنده این کلاس سطر،ستون و نام نقشه را دریافت می کند.تمام خانه های نقشه به صورت پیش فرض ، باید از نوع clear باشد.

تابعeditBlock ، خانه ی مشخص شده را به نوع تعیین شده تغییر می دهد.

تابع reSetMap() کل نقشه را و تابع reSetMap(int row,int col)خانه ی مشخص شده را به حالت پیش فرض(Block های Clear) تبدیل می کند.تابع toSting() این کلاس کل نقشه را به صورت String برمی گرداند.[[12]](#footnote-12)

1. کلاس بازی:

Game

map:Map -

airCrafts[]:AirCraft -

Game(AirCraft airCrafts[],Map map) +

function(char func):bool +

checkMoveAbility(char dir,int option):bool -

killAirCraft(AirCraft target) -

gemap():Map +

getAirCraftList():AirCraft[] +

toString() +

این کلاس چرخش نوبت ، حذف شدن بازیکن ، تغییرات در نقشه و ... را مدیریت میکند.

برای شروع بازی باید یک شی از این کلاس ساخته شود تا بتوان بازی جدید را شروع کرد.

نقشه بازی و هواپیماهای موجود در آن به وسیله ی سازنده کلاس دریافت می شوند.

در این کلاس هر هواپیما در نوبت خود ، می تواند حرکت و یا شلیک کند.تابع function ، بعد از دریافت عملگر(که در آرگومان این تابع مشخص می شود.)امکان انجام آن را بررسی میکند و در صورت موفقیت آمیز بودن ، مقدارtrue بر میگرداند.

تابع functionبرای بررسی امکان حرکت ، از تابع checkMoveAbility استفاده می کند.(حرکت غیر مجاز ، خروج از محدوده نقشه و یا برخورد به کوه می تواند باشد.)تابع toString() ، نقشه بازی ، محل و جهت هواپیما ها را نمایش میدهد.[[13]](#footnote-13)

لیست عملگر ها در تابع function:

* u:UP-حرکت هواپیما ، یک خانه به سمت بالا
* d:DOWN-حرکت هواپیما ، یک خانه به پایین
* r:RIGHT-حرکت هواپیما ، یک خانه به سمت راست
* l:LEFT-حرکت هواپیما ، یک خانه به سمت چپ
* F:FIRE-شلیک سلاح هواپیما (آرگومان option نوع سلاح را مشخص میکند.)

1. در متن راهنما، برخی از پانویس ها نمونه ای از خلاقیت هایی است که شما می توانید به پروژه اضافه کنید. [↑](#footnote-ref-1)
2. قطعاً به یک ملیت می شود چندین هواپیما اضافه کرد. ولی به یاد داشته باشید، هر بازیکن یک هواپیما را کنترل می کند. [↑](#footnote-ref-2)
3. نیازی به پیاده سازی دقیق قوانین سقوط آزاد نیست. افتادن بمب در خانه بعدی کافی است. [↑](#footnote-ref-3)
4. استفاده از شماره به همراه اسم کمی کار شما را راحت تر میکند. [↑](#footnote-ref-4)
5. سلاح های موجود در هواپیما می تواند محدود باشد. [↑](#footnote-ref-5)
6. 3 در واقعیت ، هواپیما آسیب می بیند ولی تا وقتی به طور کامل تخریب نشده می تواند پرواز کند. پس بعد از اصابت موشک ، مقداری از سلامت هواپیما کم میشود. وقتی سلامت هواپیما به 0 برسد ، هواپیما سقوط میکند. [↑](#footnote-ref-6)
7. گزارش امتیاز های کسب شده ، می تواند مهم باشد. هواپیمایی که فقط در آخرین درگیری شرکت کند و فقط یک هواپیما را از بین ببرد ، نباید برنده بازی باشد! [↑](#footnote-ref-7)
8. هر خانه می تواند قبرستان هم داشته باشد و اسم فردی که در آن خانه حذف شده را ثبت کند. [↑](#footnote-ref-8)
9. انتخاب جهت تصادفی برای خروج از مه. [↑](#footnote-ref-9)
10. می توان با وارد شدن به این خانه آن را تبدیل به خانه ی clear کرد.دستورٍ فرود از طرف هواپیما به کشتی هوایی مقدور است. [↑](#footnote-ref-10)
11. شاید بهتر باشد کمی در تابع تصادف دخالت کنید و احتمال خانه ی clear را بیشتر کنید. [↑](#footnote-ref-11)
12. وجود خانه هایی که امتیاز های ویژه مثل شلیک یا حرکت مجدد می دهد ، می تواند جالب باشد. برای مثال برخی از خانه های ضد هوایی ، امتیاز شلیک مجدد داشته باشند. [↑](#footnote-ref-12)
13. در کلاس بازی درجه های سختی تعیین کنید به طوری که هرچه درجه ی سختی بالاتر باشد خانه Random موانع بیشتری بر سر راه بازی کننده ها قرار دهد. [↑](#footnote-ref-13)